

REMARKS/ARGUMENTS

Claims 1-3, 6-10, and 18 are pending in the application. Claims 1, 2, 6 ,7, and 18 have been amended. Claims 4, 5, and 11-17 were previously canceled. Claim 21 is newly presented. No new matter has been entered through this amendment.

Claim objections

The claims were amended to address informalities. Specifically, the following changes were made to the claims for clarity and consistency:

Claim 1: "software layers" was replaced with "software stack layers" and
"being" was replaced with "is"

Claim 2: "network stack layer" was replaced with "software stack layer"

Claim 18: "software layers" was replaced with "software stack layers" and
"a memory address pointers" was replaced with "memory address pointers"

In view of these amendments, the Applicants respectfully request the Examiner to withdraw these objections.

Rejections under 35 U.S.C. § 112

Claims 1-3 were rejected under 35 U.S.C. 112, first paragraph. Examiner stated that Claim 1 was rejected because the network stack interface does not comprise a first software stack layer as recited in the claims. Claim 1 was amended so that a "first software stack layer" is now replaced with a "target software layer". This change is supported in the Applicants' disclosure in paragraph [0043], line 3.

An amendment to claim 3 was necessitated by the aforementioned amendment to claim 1. Claim 3 was amended to add the feature of a selected one of the plurality of buffer descriptors each buffer descriptor. Support for this can be found in paragraph [0018], line 3.

Applicants respectfully request the Examiner to remove the rejection of claim 1-3, in light of the amendment.

Examiner stated that Claim 1 recites "the buffer descriptor being one of a plurality of buffer descriptors that defines the data that is common to the plurality of buffer descriptors, and the plurality of buffer descriptors define transport layer header data". The Examiner further stated that none of the buffer descriptors defining the data can be considered as a buffer descriptor defining transport layer header data, and that the transport layer header data does not define data that is common to the plurality of buffer descriptors. Claim 1 has been amended to recite language found in the as-filed application. Specifically, the following feature was added: a selected one of the plurality of buffer descriptors stores a memory address and length of a buffer and passes on the memory address and length of the buffer to a next selected one of the plurality of buffer descriptors. Support for this feature this amendment can be found in paragraph [0018], line 5.

Applicants respectfully request the Examiner to remove the rejection of claim 1, 2 and 3, in light of the discussion above.

Claims 6-10 were rejected under 35 U.S.C. 112, second paragraph. Applicants respectfully traverse as explained below.

Examiner stated that Claims 6-10 depend either directly or indirectly on claim 5, which was cancelled and therefore, the scope of claims 6-10 cannot be clearly determined. In light of Examiner's remarks, Applicants have added claim 21 and have amended claims 6 and 7 so that no claims now either directly or indirectly depend from claim 5. Support for this amendment can be found at paragraph 41, line 4. Applicants respectfully request the Examiner to withdraw the rejection of claim 5, in light of the aforementioned amendment..

Regarding the Examiner's note made in comment 8., Examiner stated that "claim 1 recites 'data' and 'transport layer header data', which are interpreted to be two different types of data - as there is no recitation in the claim for the data to only include transport layer header data." This point is now moot as claim 1 has been amended, and no longer recites "transport layer header data".

Rejections under 35 U.S.C. § 102

Claims 1-3 and 18 were rejected under 35 U.S.C. 102 (b) as being anticipated by Aditya et al. (US 5,729,681).

Claim 1 and 18 have been amended to better define that the network stack interface is defined to communicate between the transport layer and any higher layer within the network stack. Applicants define a network stack interface that includes a header portion and a buffer descriptor. The buffer descriptor includes a memory address pointer to the data. (pg. 6, paragraph [0017]). Aditya does not teach this, but instead teaches that Transmit Control Blocks (TCB) with their associated fragment structures are used to transfer data from the lower protocol layer 36 via a driver layer 60, to the controller (column 2, line 54-58, and Fig. 2). Aditya teaches a layered architecture system which is shown in the block diagram of Fig. 2. Figure 2 shows a typical network operating system (NOS) protocol stack (the stack). The top layer of the stack is the application layer 32, followed next by an upper protocol layer 34, then a lower protocol layer 36, and a driver layer 60. Also included in the NOS protocol stack (but not shown) is the physical layer (the lowest layer of the stack) consisting of the controller 14 and the network physical link 24 (col. 2, line 12-16). Aditya, therefore, teaches that the transfer of data using Trans Buffer Address 64, 66, 68 by the TCB (and fragment structures) is occurring from the lower protocol layer down to the next lower level (the driver layer) down to the controller which is in the physical layer, i.e. the lowest level of the stack.

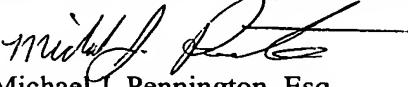
This is in contrast to the Applicants' network stack interface, which uses buffer descriptors that reside in the transport layer, to pass memory address pointers (which are part of the buffer descriptors) between software stack layers between the transport layer and any higher level in the software stack layer. That is, pointers are passed up to the STP layer 56 (transport layer), using a SID which includes a buffer descriptor 104 (pg. 13, paragraph [0049]), and then passed to the SEP layer 54, and finally the SCSI target layer 52. Although Aditya's pointers 65, 67, and 69 identify buffers in layer 36, the logic for this pointing is not in the transport layer, as now claimed.

The Open Systems Interconnection (OSI) model is made up of 7 functional layers (Fig. 1). The highest layer is the application layer 16, followed by a presentation layer 18, a session layer 20, a transport layer 22, a network layer 24, a data link layer 26, and a physical layer 28. As explained in the previous paragraph, Applicants claim the memory address pointer is passed between software stack layers via the network stack interface that is defined between the transport layer and any higher layer. As shown in the OSI reference model of Fig. 1, the physical layer is below the transport layer.

Applicants respectfully request the Examiner to remove the rejection of claims 1-3, and 18, in view of the foregoing.

In view of the foregoing, Applicants respectfully submit that all of the pending claims are in condition for allowance. A notice of allowance is respectfully requested. In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 774-6927. If any fees are due in connection with the filing of this paper, then the Commissioner is authorized to charge such fees to Deposit Account No. 50-0805 (Order No. ADAPP166A). A copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, L.L.P.


Michael J. Pennington, Esq.
Reg. No. 59,064

MARTINE PENILLA & GENCARELLA, L.L.P.
710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
Telephone: (408) 749-6900
Customer Number 25920